**Faculdade de Engenharia da Universidade do Porto**



**Informações sobre Hotéis em Linguagem Natural**

*Relatório Final*

**Inteligência Artificial 3ºAno**

**Mestrado Integrado em Engenharia Informática e Computação**

**29 de maio de 2016**

**Estudantes:**

Gil Manuel Oliveira de Almeida Domingues, [up201304646@fe.up.pt](mailto:up201304646@fe.up.pt)

Pedro Martins Pontes, [up201305367@fe.up.pt](mailto:up201305367@fe.up.pt)

Resumo

No presente relatório – elaborado no âmbito da unidade curricular Inteligência Artificial – detalha-se a implementação de um programa que permite obter e definir informações sobre hotéis, usando linguagem natural, como se de um balcão de informações se tratasse.

Palavras-Chave

*Definite Clause Grammars, HTML, JavaScript, Linguagem Natural, Prolog, Representação do Conhecimento.*

Índice

[1. Introdução 5](#_Toc452132475)

[2. Especificação 5](#_Toc452132476)

[3. Implementação 5](#_Toc452132477)

[3.1. Estrutura 5](#_Toc452132478)

[3.2. Desenvolvimento 7](#_Toc452132479)

[3.3. Detalhes 7](#_Toc452132480)

[4. Avaliação e Validação 8](#_Toc452132481)

[5. Conclusões 10](#_Toc452132482)

[6. Recursos 10](#_Toc452132483)

[6.1. Bibliografia 10](#_Toc452132484)

[6.2. Contribuições 10](#_Toc452132485)

[7. Anexos 11](#_Toc452132486)

[A. Utilização 11](#_Toc452132487)

## Introdução

O turismo, congregando em si um alargado conjunto de atividades e produtos de diversa natureza, desempenha um papel de crescente relevância para a economia nacional. Desde logo, existe uma pluralidade de meios para divulgação de informações, como postos de turismo, revistas, *websites* e *apps*.

Com este trabalho pretendia implementar-se um programa que permita obter informações sobre hotéis, usando linguagem natural, como se de um balcão de informações se tratasse.

Ao longo do presente relatório estabelecem-se objetivos, procede-se à especificação do programa e à análise do trabalho desenvolvido. Por último, abordam-se questões de avaliação e validação do produto final.

# Especificação

O programa desenvolvido faz uso de uma base de conhecimento contendo informações acerca de hotéis – designadamente, localização, categoria, serviços disponíveis, quartos e respetivos preços.

O acesso a esta informação é feito por meio de interrogações e afirmações – escritas em português –, sendo ainda possível adicionar informação a essa base de conhecimento. O programa interpreta e responde a frases escritas em linguagem natural usando uma *DCG* – *Definite* *Clause* *Grammar.*

O programa começa por validar (ou invalidar) as afirmações e interrogações, verificando a sua sintaxe, bem como a sua semântica.

A inclusão de mais que um comando numa mesma frase é traduzida pelo uso da partícula "e" nessa frase, e o programa guarda o contexto da frase anterior para permitir dar resposta a interrogações iniciadas pela partícula “e”.

1. Implementação
   1. Estrutura

Para a implementação, considerou-se a divisão do programa em cinco módulos:

* um módulo com a Representação de Conhecimento, contendo a informação do sistema;
* um módulo com o léxico reconhecido pelo programa;
* um módulo responsável pela análise das frases;
* um módulo responsável pela geração das respostas;
* um módulo de interface de utilizador.

E:\Transferências\components.png

Figura 1 - Estruturação do programa em módulos.

Com base nesta divisão, definiram-se as seguintes etapas:

* Definição de uma base de conhecimento

Esta etapa compreendeu a definição de uma base de conhecimento e o seu preenchimento com informação acerca de hotéis.

* Implementação da análise lexical e sintática

A definição do léxico com vocabulário relacionado com o tema foi fundamental para testar frases de acordo com uma gramática.

Nesta fase do trabalho procedeu-se também à implementação da analise sintática, por modo a validar uma questão ou afirmação, e implicou a criação de um conjunto de regras de uma gramática.

Essa gramática será definida como um *DCG* – *Definitve* *Clause* *Grammar*, que permite estabelecer um conjunto de regras recursivas que permitem validar todas as frases possíveis de escrever com o léxico definido e respeitando as regras fundamentais da concordância da gramática da língua portuguesa.

* Implementação da análise semântica e lógica

Nesta etapa procedeu-se à interpretação da frase e à sua tradução num comando de pesquisa apropriado. Esta fase incluiu ainda a avaliação desse comando, produzindo uma resposta.

* Implementação da interface do utilizador

Nesta etapa final, procedeu-se à implementação de uma interface HTML para interagir com o programa, que faz uso de *sockets* para enviar pedidos ao programa, que, agindo como um servidor, analisa a frase, tentando encontrar uma resposta para o pedido.

* 1. Desenvolvimento

O programa foi desenvolvido em ambiente *Windows* *10*, recorrendo ao editor de texto *Atom.* Para executar o código desenvolvido, recorreu-se ao S*ICStus Prolog*, e o *browser* em que a interface foi testada foi o *Google Chrome*.

A implementação de todos os módulos definidos foi feita usando unicamente *Prolog* como linguagem de programação, com exceção do módulo de interface com o utilizador, em que se recorreu também a *HTML, CSS* e *JavaScript.*

* 1. Detalhes

No que respeita à base de conhecimento e à sua representação, foi desenvolvido um esquema composto pelos seguintes predicados:

servico(?Id, ?Nome).

localização(?Id, ?Nome).

hotel(?Nome, ?Localizacao, ?Categoria, ?Servicos, ?Quartos).

O acesso a essa base de conhecimento pode ser feito usando os predicados:

adicionar\_hotel(+Nome,+Localizacao, +Estrelas).

classificar\_hotel(+Nome, +Estrelas).

adicionar\_servicos\_hotel(+Nome, +Servicos).

adicionar\_quartos(+Nome, +Quartos).

obter\_servicos(+ServicoIds, -Servicos).

disponibiliza(+Nome, +Servicos).

quarto(+Nome,+Quartos).

Assim, definiu-se uma base de conhecimento, com informação sobre alguns hotéis, havendo a possibilidade de lhe acrescentar informação.

Na definição da *DCG*, considerou-se que uma frase pode ser de três tipos: uma interrogação, uma afirmação ou uma declaração. Uma vez validada a frase, procede-se à análise do pedido, de acordo com o tipo de frase:

processar(+Frase, -Resposta):-

frase(TipoF, TipoQ, Q, Frase, []),

analisar(TipoF, TipoQ, Q, Resposta).

**Interrogações**

frase(interrogacao, TipoQ, -Q) - -> frase\_int(TipoQ, Q), ([?]; []).

analisar(interrogacao, +TipoQ, +[At, QIn], -Resposta):-

prepare\_query(QIn, QOut), !,

(TipoQ=contexto, !,

retomar\_contexto(QOut, TipoQ2, At, Q);

novo\_contexto(TipoQ, At, QOut), Q=QOut, TipoQ2=TipoQ),

responder(TipoQ2, At, Q, Resposta).

Por modo a suportar interrogações em contexto, definiu-se um predicado dinâmico para guardar informação contextual, que pode ser usada pela questão a seguir:

novo\_contexto(?TipoQuestao, ?Atributo,?NovoContexto).

retomar\_contexto(+NovoContexto, +TipoQuestao, +Atributo, -ContextoAtual).

**Afirmações**

frase(afirmacao, \_, -Q) - -> frase\_afir([], Q), ([.]; [?]; []).

analisar(afirmacao, \_, +QIn, -Resposta):-

prepare\_query(QIn, QOut), !,

concordar(QOut, Resposta).

**Declarações**

frase(declaracao, \_, -Q) - -> frase\_afir([], Q), [;].

analisar(declaracao, \_, +QIn, -Resposta):-

prepare\_query(QIn, QOut),!,

memorizar(QOut, Resposta).

1. Avaliação e Validação

A avaliação do correto funcionamento do programa passou pela utilização de diversos casos de teste. Seguidamente, listam-se alguns desses casos, seguidos pela resposta esperada, em função dos dados inicialmente presentes na base de conhecimento do programa.

**Afirmações**

* O hotel Moliceiro fica em Espanha e possui 4 estrelas.
  + Erro gramatical.
    - Exemplo de erro gramatical: ‘Espanha’ não consta do léxico definido.
* O hotel Moliceiro fica em Porto.
  + Erro gramatical.
    - Exemplo de erro gramatical: A palavra ‘Porto’ requer a preposição ‘no’ ou ‘do’.
* O hotel Veneza fica em Aveiro e possui 4 estrelas.
  + Errado…
* O hotel Intercontinental disponibiliza o serviço de babysitting e de wi-fi.
  + Sim!

**Interrogações**

* Quantos hotéis de Lisboa possuem categoria inferior a 4 e quartos executivos?
  + 1
* Quais os hotéis de categoria superior a 3 estrelas em Lisboa?
  + [Corinthia]
* E em Aveiro?
  + [Moliceiro]
    - Exemplo de questão usando contexto, introduzido pela partícula ‘e’.
* Que serviços disponibiliza o hotel Eduardo VIII?
  + [serviço-de-quartos, bar, estacionamento, wi-fi, pequeno-almoço, ar-condicionado]
* Quais os hotéis portuenses que possuem serviço de *babysitting*?
  + [Intercontinental]
* Quais os hotéis de Viana que possuem categoria igual a 4 e quartos com vista de mar?
  + [Flor de Sal]

**Declarações**

* O Hotel Ferreiro fica em Aveiro e tem 3 estrelas;
  + Novo hotel criado.
    - Exemplo de declaração de nova informação. Realçar a necessidade de terminar a frase com ‘;’, sob pena de a declaração ser tomada por uma afirmação.
* O Hotel Ferreiro tem 1 estrela;
  + Categoria atualizada.

1. Conclusões

Analisado o trabalho desenvolvido, consideram-se atingidos os objetivos.

Definiu-se uma base de conhecimento com informação sobre vários hotéis, – designadamente, localização, categoria, serviços disponíveis, quartos e respetivos preços.

Em seguida, especificou-se um léxico adequado e implementou-se uma *DCG* com vista à análise de declarações, afirmações e interrogações.

Adicionalmente, construiu-se uma interface *web* para interagir com o programa.

A nível de trabalho futuro, uma melhoria possível passa por aumentar flexibilidade em relação à sintaxe aceite, tendo em mente que o português é um idioma de elevada complexidade. Seria igualmente interessante implementar respostas em linguagem natural – possivelmente, tentando corrigir o utilizador –, por forma a tornar a interação com o sistema mais amigável.

1. Recursos
   1. Bibliografia

[1] Oliveira, Eugénio. *Capítulo III – Introdução à Representação do Conhecimento*.

Universidade do Porto [Consultado a 1-04-2016 – 17-04-2016]. Disponível em: <https://web.fe.up.pt/~eol/IA/1516/APONTAMENTOS/3a_EC.pdf>.

[2] Oliveira, Eugénio. *Capítulo V – Linguagem Natural Computacional*.

Universidade do Porto [Consultado a 4-04-2016 – 17-04-2016]. Disponível em: <https://web.fe.up.pt/~eol/IA/1516/APONTAMENTOS/5_Ln.pdf>.

* 1. Contribuições
* Gil Domingues: 40%
* Pedro Pontes: 60%

Anexos

1. Utilização

O programa pode ser utilizado exclusivamente a partir da interface do S*ICStus Prolog* ou a partir de uma interface *web*.

1. Consultar o ficheiro sobre\_hoteis.pl;

***SICStus Prolog***

1. Introduzir o comando:

processar(+Frase, -Resposta).

onde Frase representa uma lista com as palavras da frase.

**Advertência**

Por motivos operacionais, todas as palavras do léxico encontram-se em letra minúscula e sem acentuação. Em particular, a forma verbal ‘é’ encontra-se definida como ‘e\_’, por forma a distingui-la da conjunção ‘e’. Em caso de dúvida, consultar o ficheiro “léxico.pl” para.

**Interface *web***

1. Introduzir, sem argumentos, o comando:

servidor.

1. Abrir o ficheiro index.html;